Efficient Packet Classification Algorithm Based on Entropy

Michal Kajan, Jan Kořenek
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, Brno, Czech Republic
{ikajan,korenek}@fit.vutbr.cz

ABSTRACT

This paper deals with packet classification in high-speed networks. It introduces a novel method for packet classification based on the amount of information stored in the ruleset. Basic principles of the algorithm based on the effort to reduce the amount of the necessary memory space and number of computational steps are presented together with analysis of the input rulesets.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection (e.g., firewalls)*

General Terms

Design, Performance, Security, Complexity

Keywords

Packet Classification, Algorithms, Entropy

1. INTRODUCTION

The great advance in Internet technologies and available variety of network services have caused an increasing demand in number of connected users. Internet has turned into an important infrastructure utilized for commercial purposes. Obtainable network speeds reach rates of hundreds of Mb/s for end users, for service providers at their backbones reach up to tens of Gb/s.

No wonder, such a development breeds requierements for providing desired performance and security in computer networks. Packet classification is one of the basic steps in ensuring security demands. With a predefined set of rules, packet classifier is performing decisions about discarding or forwarding the incoming packet. However, packet classifier is not only intended to provide basic filtering using input rulesets, with advantage it can be used for traffic shaping and traffic policing, ensuring reliable function of virtual private networks, lawful interception operations, etc.

2. ALGORITHM

There has been a great research effort in the field of packet classification in the last decade. From the existing approaches we can consider two groups of algorithms to be most popular — crossproduct based (decomposition based) and decision-tree based (space-division based). The representatives of the former group are the MSCA [1], the DCFL [6] and the PHCA [3] algorithm, and of the latter group the HiCuts [2] and the HyperCuts [4] algorithm.

Decomposition based algorithms perform classification in several steps. In general, first step is the longest prefix match of the value for each packet header field followed by a method in the second step which processes the results of the prefix match and determines the matching rule. Second step is ussually a combination (thus *crossproduct* [5]) of the prefix matching results.

Second group constitutes of algorithms creating the decision tree during the phase of the ruleset preprocessing. These algorithms divide the n-dimensional space (created by the input ruleset with each rule specifying several fields) into smaller subspaces. Rules represent n-dimensional figure and the aim is is to find the smallest geometric figure containig the packet (represented as a point in this space).

We can further observe a character of the input rules, analyze and use the techniques from previously mentioned approaches to construct a new classification algorithm. Basically, each rule can be represented as a sequence of bits. The amount of information stored can be analyzed not only troughout each field (dimension) but also in individual bit columns separately. According to such an analysis we are able to construct a big decision tree representing the space of all possible results in classification. The idea of tree construction is in the figure 1.

Each bit column is sorted according the amount of information. Note that the resulting ordering does not take into account the dimensions. Thus, the bit columns may be completely mixed together. Higher parts of the tree located near the root node will be denser than the lower parts.

The goal of this approach is to achieve superior classification speed at least comparable to existing solutions together with acceptable memory requierements. The traversal of the tree in the higher parts can be done using small number of bits in one computational step due to the effectivity. Nevertheless, in the lower parts it could be possible to process high number

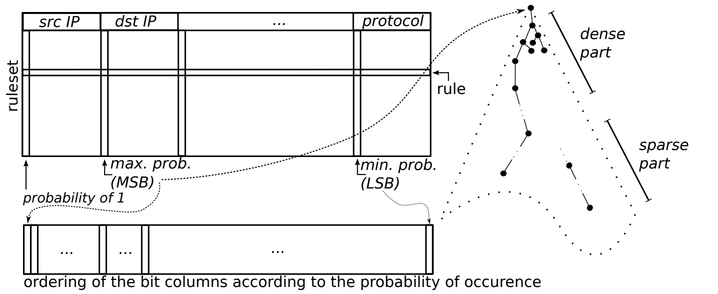

Figure 1: Tree construction using an amount of information

of bits using hash functions to save the available memory.

We have analyzed several synthetic (generated by Classbench [7]) and real rulesets. Whole ruleset can be considered as a table, where a row represents one specific rule and a column is a bit column of a specific header field. For each bit column the probability of a bit value 1 was computed. The term probability refers to the frequency occurence of such a value with the respect to the total number of rows (rules). Moreover, the rules also specify fields (or their parts) with don't care bits. As it is not important which value will have such a bit in the packet header, they were assigned value 0.5 for the occurence computation. The results can be seen in the figures 2 and 3. The X-axis represent the bit column, the Y-axis the probability of occurence of bit value 1. Analysis is based on the average values from several rulesets.

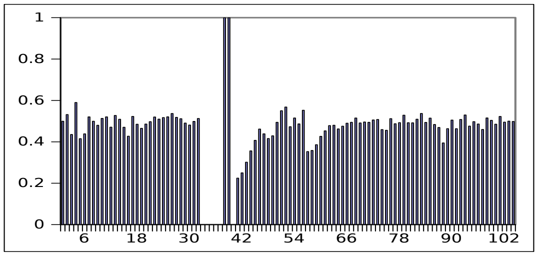

Figure 2: Synthetic rulesets analysis

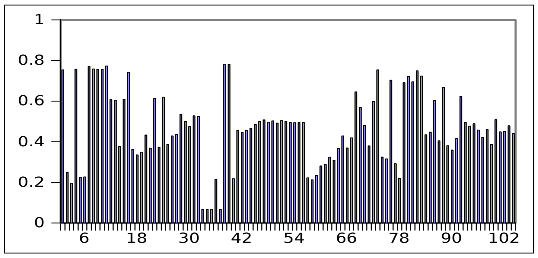

Figure 3: Real rulesets analysis

For each rule 5 fields were analysed, bits from fields are ordered from MSB and in the figures are in this order: source IP address (v4, 32b), protocol value (8b), source and destination ports (16b+16b) and destination IP address (v4, 32b) which results to 104 bit columns in total.

The distribution of values in the synthetic rulesets is very close to the uniform distribution. On the other hand, the distribution for the real rulesets is much more interesting from the point of resulting bit ordering for the construction of the tree. Note the peak values for protocol bits (33-40), this is due the fact that all rules in the synthetic rulesets specify TCP protocol as the only value, in the real rulesets there are other protocols specified as well.

3. FURTHER WORK

In the previous section we have denoted the novel concept of packet classification technique based on entropy. The mentioned technique does need to perform additional analyses and experiments for final design of the classification process. It will be necessary to determine the proper method for tree traversal in some parts of the tree to reach the lower parts and consequent computational process e.g. with hash functions for quick traversal to leaf nodes and determination of the matching rule. To conclude, we believe this approach appears to be promising from the point of the necessary storage needed for correct classification together with high achievable rates. These attributes make it applicable for practical use in the area of high-speed packet classification and competitive to other solutions.

Acknowledgment

This research has been partially supported by the Research Plan No. MSM, 0021630528 – Security-Oriented Research in Information Technology and the grant BUT FIT-S-10-1.

4. REFERENCES

[1] S. Dharmapurikar, H. Song, J. Turner, and J. Lockwood. Fast packet classification using Bloom filters. In *ANCS '06: Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*, pages 61–70, New York, NY, USA, 2006. ACM.

[2] P. Gupta and N. McKeown. Packet classification using hierarchical intelligent cuttings. In *Proc. Hot Interconnects*, 1999.

[3] V. Puš and J. Kořenek. Fast and scalable packet classification using perfect hash functions. In *FPGA '09: Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays*, pages 229–236, New York, NY, USA, 2009. ACM.

[4] S. Singh, F. Baboescu, G. Varghese, and J. Wang. Packet classification using multidimensional cutting. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 213–224, New York, NY, USA, 2003. ACM.

[5] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel. Fast and scalable layer four switching. *SIGCOMM Comput. Commun. Rev.*, 28(4):191–202, 1998.

[6] D. Taylor and J. Turner. Scalable packet classification using distributed crossproducing of field labels. In *IEEE INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, pages 269–280, July 2005.

[7] D. E. Taylor and J. S. Turner. Classbench: a packet classification benchmark. *IEEE/ACM Trans. Netw.*, 15(3):499–511, 2007.